

SEGMENTASI BERBASIS REGION PADA CITRA BERWARNA UNTUK KEPERLUAN TEMU KEMBALI CITRA PADA EVENT OLAH RAGA LAPANGAN HIJAU

Arif Basofi, S.Kom¹, Moch. Hariadi, ST, M.Sc, Ph.D.²

Program Magister
Bidang Keahlian Jaringan Cerdas Multimedia
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 60111, Indonesia
E-mail : ¹ariv@eepis-its.edu, ²mochar@ee.its.ac.id

Abstrak

Dalam sistem temu kembali citra, segmentasi adalah bagian terpenting dalam tahap awal pemrosesan citra. Metode segmentasi yang tepat sangat mempengaruhi hasil segmentasi suatu citra, terutama pada citra berwarna dan bertekstur yang menjadi permasalahan tersendiri dalam proses segmentasi. Salah satu metode yang diterapkan pada penelitian ini adalah segmentasi berbasis region dengan algoritma JSEG.

Proses segmentasi citra dengan algoritma JSEG terbagi atas dua tahap, yaitu proses kuantisasi warna dan proses segmentasi spasial. Penggunaan ruang warna CIE LUV pada tahap kuantisasi warna yang mampu menerima warna menurut persepsi manusia. Warna-warna citra yang terkuantisasi membentuk color class-map untuk membedakan region-region dalam citra. Di tahap segmentasi spasial, dilakukan perhitungan ukuran segmentasi yang "baik" menurut color class-map yang terbentuk pada window lokalnya sehingga menghasilkan "J-image". Metode region growing digunakan untuk mensegmentasi citra berdasarkan J-image multiskala.

Uji coba pada 12 citra event olah raga lapangan hijau dilakukan untuk melihat hasil segmentasi yang sesuai menurut kombinasi nilai parameter threshold yang tepat. Dari variasi parameter threshold kuantisasi warna diperoleh 67% cenderung pada nilai threshold 255 menghasilkan segmentasi yang baik. Sedangkan untuk threshold region merging, cenderung pada nilai threshold 0.4. Hasil eksperimen menunjukkan bahwa dengan kombinasi parameter nilai threshold yang baik ini, memudahkan dalam proses implementasi sistem temu kembali citra menurut hasil segmentasi warna.

Kata kunci : segmentasi citra berwarna, JSEG, CIE LUV, region growing, region merging

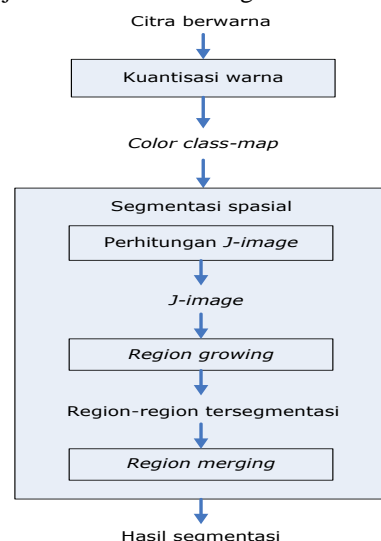
1 Pendahuluan

Segmentasi citra berwarna sangat bermanfaat terutama dalam aplikasi temu kembali citra.

Permasalahan yang sering muncul disebabkan tekstur citra, jika citra hanya terdiri dari region-region warna yang homogen, metode *clustering* dalam ruang warna dapat diterapkan untuk segmentasi citra [1]. Namun pada kenyataannya, citra pemandangan alam kaya akan warna dan tekstur sehingga sulit untuk mengidentifikasi region-region pada citra yang mengandung pola-pola warna-tekstur. Pendekatan yang digunakan dalam penelitian ini berdasarkan pada asumsi-asumsi sebagai berikut :

- Setiap region pada citra mengandung pola warna-tekstur yang terdistribusi *uniform*,
- Informasi warna pada setiap region citra dapat dinyatakan dengan beberapa warna-warna terkuantisasi, yang tepat untuk sebagian besar citra pemandangan alam berwarna,
- Warna-warna antara dua region tetangga dapat dibedakan. Hal ini merupakan asumsi dasar dari semua algoritma segmentasi citra berwarna.

Algoritma JSEG memerlukan tiga parameter, pertama, threshold untuk proses kuantisasi warna. Threshold tersebut menentukan jarak minimum antara dua *cluster* warna yang akan digabungkan. Kedua, jumlah skala segmentasi dan ketiga, threshold untuk *region merging*. Gambar 1 menunjukkan Blok Sistem algoritma JSEG [2].



Gambar 1. Blok Sistem Algoritma JSEG

2 Metode

2.1 Kuantisasi Warna

Pertama, warna-warna pada citra dikuantisasi secara kasar tanpa menurunkan kualitas warna secara signifikan. Tujuannya adalah untuk mengekstrak beberapa perwakilan warna yang dapat digunakan untuk membedakan region-region dalam citra. Secara khusus, diperlukan 10 sampai 20 warna dalam citra pemandangan alam. Kuantisasi warna yang baik berpengaruh terhadap proses segmentasi. Dalam implementasi penelitian ini digunakan algoritma kuantisasi warna perseptual [3].

Algoritma kuantisasi warna perseptual bekerja berdasarkan persepsi penglihatan manusia yang lebih sensitif terhadap perubahan pada *smooth region* daripada perubahan pada *detailed region* (*textured region*). *Smooth region* adalah *boundary region-region* yang memungkinkan, sedangkan *detailed region* adalah region-region homogen pada citra. Karena itu, warna-warna dapat dikuantisasi secara lebih kasar pada *detailed region* tanpa mempengaruhi kualitas perseptual secara signifikan. Berdasarkan fakta tersebut, setiap piksel ditandai dengan bobot yang berdasarkan pada *variance* dalam window lokal sedemikian rupa sehingga piksel-piksel pada *smooth region* lebih penting daripada piksel-piksel pada *detailed region*.

Algoritma ini menggunakan statistik lokal yang diperoleh setelah *peer group filtering*, yaitu bobot dalam proses kuantisasi vektor. Prosedur kuantisasi warna adalah sebagai berikut :

1. Konversi ruang warna RGB ke LUV untuk menjaga kualitas warna.
2. Pertama, *peer group filtering* diterapkan untuk menghaluskan citra dan menghilangkan *impulse noise*. Hasilnya berupa :
 - x : vektor piksel citra yang telah dihaluskan oleh anggota *peer group*-nya
 - N : jumlah *cluster* awal
 - v : bobot perseptual untuk setiap piksel
3. *Clustering* dengan *Generalized Lloyd Algorithm* (GLA). Hasilnya berupa centroid untuk setiap *cluster* warna.
4. Penggabungan *cluster-cluster* yang jarak centroidnya kurang dari threshold kuantisasi warna.
5. Klasifikasi piksel ke dalam *cluster* warna yang centroidnya terdekat dengan intensitas piksel tersebut.
6. Konversi ruang warna LUV ke RGB untuk menampilkan citra hasil kuantisasi warna.

2.1.1 Peer Group Filtering (PGF) [3]

Misalkan $x_0(n)$ menyatakan vektor piksel citra yang memberikan ciri informasi warna pada posisi n yang berpusat pada window $w \times w$. Urutkan semua piksel pada window tersebut berdasarkan jaraknya terhadap $x_0(n)$ dalam urutan *ascending*

dan nyatakan sebagai $x_i(n)$, $i = 0, \dots, k = w^2 - 1$. Ukuran jarak Euclidean yang digunakan adalah :

$$d_i(n) = \|x_0(n) - x_i(n)\|, i = 0, \dots, k \quad (1)$$

$$d_0(n) \leq d_1(n) \leq \dots \leq d_k(n) \quad (2)$$

Peer group $P(m, w)$ untuk $x_0(n)$ terdiri dari m piksel yang intensitasnya terdekat dengan $x_0(n)$ dalam window $w \times w$ yang berpusat pada $x_0(n)$. Penentuan m menggunakan estimasi diskriminan Fisher. *Criterion* yang dimaksimalkan adalah :

$$J(i) = \frac{|a_1(i) - a_2(i)|^2}{s_1^2(i) + s_2^2(i)}, i = 1, \dots, k \quad (3)$$

di mana :

$$a_1(i) = \frac{1}{i} \sum_{j=0}^{i-1} d_j(n), a_2(i) = \frac{1}{k+1-i} \sum_{j=i}^k d_j(n) \quad (4)$$

$$s_1^2(i) = \sum_{j=0}^{i-1} |d_j(n) - a_1(i)|^2, s_2^2(i) = \sum_{j=i}^k |d_j(n) - a_2(i)|^2 \quad (5)$$

m adalah indeks di mana $J(i)$ bernilai maksimum. Selanjutnya $x_0(n)$ diganti dengan rata-rata anggota *peer group*-nya.

Untuk menghilangkan efek *impulse noise*, turunan pertama dari jarak $d_i(n)$, $f_i(n)$, dihitung sebelum klasifikasi *peer group* :

$$f_i(n) = d_{i+1}(n) - d_i(n) \quad (6)$$

Pengujian dilakukan terhadap M titik pertama dan terakhir dari $x_i(n)$ untuk memeriksa apakah titik-titik tersebut termasuk *impulse noise* :

$$f_i(n) \leq \alpha \quad (7)$$

di mana $M = w / 2$, separuh dari ukuran window, dan α diset bernilai tinggi untuk citra yang sangat rusak dan diset bernilai rendah untuk citra yang sedikit rusak. Jika $f_i(n)$ tidak memenuhi kondisi tersebut, maka titik-titik terakhir $x_j(n)$ untuk $j \leq i$ atau $j > i$ dianggap sebagai *impulse noise* dan dihilangkan. Kemudian $d_j(n)$ sisanya digunakan untuk mengestimasi *peer group* yang sebenarnya.

Selanjutnya dilakukan perhitungan jarak maksimum setiap *peer group* $T(n)$. Nilai $T(n)$ mengindikasikan kehalusan region lokal. Bobot perseptual untuk setiap piksel $v(n)$ dihitung dengan :

$$v(n) = \exp(-T(n)) \quad (8)$$

sehingga piksel-piksel pada *detailed region* memiliki bobot yang lebih rendah daripada piksel-piksel pada *smooth region*.

Rata-rata $T(n)$, T_{avg} , mengindikasikan kehalusan keseluruhan citra. Secara umum, semakin besar nilai T_{avg} , semakin berkurang kehalusan citra dan semakin banyak jumlah *cluster* yang diperlukan untuk kuantisasi warna. Jumlah awal *cluster* N diestimasi dengan :

$$N = \beta T_{avg} \quad (9)$$

di mana β diset bernilai 2 pada percobaan.

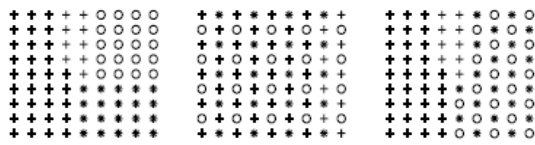
2.1.1 Generalized Lloyd Algorithm (GLA)

GLA digunakan untuk menandai piksel-piksel yang berbobot rendah dengan *cluster* yang lebih sedikit untuk mengurangi jumlah *cluster* warna pada *detailed region*. Centroid untuk *cluster* warna C_i dihitung dengan :

$$c_i = \frac{\sum v(n)x(n)}{\sum v(n)}, x(n) \in C_i \quad (10)$$

2.2 Segmentasi Spasial

Setelah tahap kuantisasi warna, diperoleh *color class-map*, yaitu label yang ditandai pada piksel-piksel dalam citra yang menyatakan klasifikasi *cluster* warna untuk piksel tersebut. Gambar 2 menunjukkan contoh *color class-map*. Nilai label diwakili oleh tiga simbol, '*', '+', dan 'o'.



Gambar 2. Contoh *color class-map*

Gambar 3 menunjukkan *flow chart* tahap segmentasi spasial. Mula-mula, citra input dianggap sebagai satu region inisial. Algoritma ini kemudian mensegmentasi semua region dalam citra pada skala inisial yang besar. Proses tersebut diulang pada region-region tersegmentasi baru pada skala berikutnya yang lebih kecil hingga mencapai skala minimum yang telah ditentukan.



Gambar 3. *Flow chart* tahap segmentasi spasial

Tabel 1 berisi himpunan skala dan ukuran region yang sesuai untuk skala tersebut. Misalnya, jika ukuran citra lebih besar daripada 256 x 256, tetapi lebih kecil daripada 512 x 512, skala awalnya adalah 3.

Tabel 1. Ukuran window pada skala yang berbeda [2]

| Skala | Window (piksel) | Sampling (1/piksel) | Ukuran Region (piksel) | Valley Minimum (piksel) |
|-------|-----------------|---------------------|------------------------|-------------------------|
| 1 | 9 x 9 | 1 / (1 x 1) | 64 x 64 | 32 |
| 2 | 17 x 17 | 1 / (2 x 2) | 128 x 128 | 128 |
| 3 | 33 x 33 | 1 / (4 x 4) | 256 x 256 | 512 |
| 4 | 65 x 65 | 1 / (8 x 8) | 512 x 512 | 2048 |

2.2.1 Perhitungan *J-Image*

J-image adalah citra *grayscale* yang nilai piksel-piksennya adalah nilai J yang dihitung terhadap window lokal yang berpusat pada piksel tersebut.

$$J = \frac{S_B}{S_W} = \frac{(S_T - S_W)}{S_W} \quad (11)$$

di mana :

$$S_T = \sum_{z \in Z} \|z - m\|^2 \quad (12)$$

dan

$$S_W = \sum_{i=1}^C S_i = \sum_{i=1}^C \sum_{z \in Z_i} \|z - m_i\|^2 \quad (13)$$

Keterangan :

S_T : jarak antar kelas-kelas warna yang berbeda (*between class scatter matrix*)

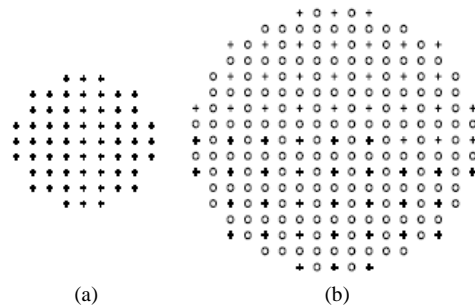
S_W : jarak antar anggota dalam setiap kelas warna (*within class scatter matrix*)

m : rata-rata lokasi spasial piksel dalam window

m_i : rata-rata lokasi spasial piksel dalam kelas warna i

Semakin besar nilai J , maka piksel tersebut semakin mendekati *boundary region*. *J-image* dapat dilihat sebagai peta yang mengandung *valley* dan *mountain* yang berturut-turut mewakili pusat region dan *boundary region*.

Window yang digunakan dalam perhitungan *J-image* berbeda-beda dalam setiap skala segmentasi, seperti pada Gambar 4. Tepian window dihilangkan untuk membuat window lebih sirkuler sehingga tidak menimbulkan bias terhadap objek segi empat.



Gambar 4. Window untuk perhitungan nilai J lokal
(a) Window dasar pada skala 1. (b) Ilustrasi window pada skala 2. Hanya titik-titik bertanda '+' yang digunakan untuk perhitungan nilai J lokal, sehingga membentuk window dasar yang sama dengan (a).

2.2.2 Region Growing

Region growing digunakan untuk mengelompokkan piksel-piksel ke dalam region-region. Hasilnya berupa *region-map*, yaitu label yang ditandai pada piksel-piksel dalam citra yang merupakan klasifikasi region untuk piksel tersebut. *Region growing* terdiri dari dua tahap, yaitu :

- *Valley determination*

Valley determination digunakan untuk menentukan himpunan *valley* terbaik. Piksel-piksel dengan nilai *J* kurang dari threshold dihubungkan untuk membentuk kandidat *valley*. Kandidat *valley* yang ukurannya melebihi ukuran *valley* minimum pada skala segmentasi yang bersesuaian (seperti pada Tabel 1) akan menjadi *valley*.

- *Valley growing*

Valley growing merupakan proses pembentukan region-region dari *valley-valley*.

2.2.3 Region Merging

Region merging digunakan untuk menggabungkan region-region yang jaraknya kurang dari threshold *region merging*. Mula-mula, dilakukan perhitungan jarak antara dua region tetangga dan hasilnya disimpan dalam tabel jarak. Kemudian pasangan region dengan jarak minimum digabungkan. Vektor fitur warna untuk region tersebut dihitung dan tabel jarak *diupdate*. Proses tersebut berlanjut hingga mencapai threshold maksimum untuk jarak. Setelah *region merging*, diperoleh hasil segmentasi.

3 Hasil Uji Coba

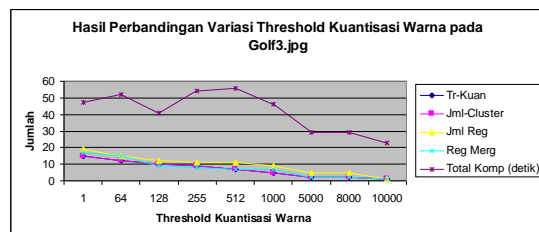
Uji coba dilakukan pada 12 citra berwarna *event* olah raga lapangan hijau, yang mencakup golf dan sepak bola dengan ukuran 128 x 128. Tujuan pengujian ini untuk melihat dan menganalisa hasil segmentasi yang baik menurut kombinasi nilai parameter yang tepat.

Uji coba dilakukan dengan variasi nilai parameter threshold kuantisasi warna pada setiap citra dengan nilai mencakup: 1, 64, 128, 512, 800, 1000, 2500, 8000 dan 10000. Untuk skala diset default 1 yang secara otomatis mengikuti ukuran dari citra. Sedangkan nilai parameter threshold untuk region merging, berkisar antara 0 hingga 1 dan diberi nilai default yang sesuai yaitu 0.4.

Hasil segmentasi pada salah satu citra *event* olah raga golf3.jpg ditunjukkan pada Gambar 5. Dengan variasi nilai parameter threshold kuantisasi, jumlah skala 1 (oleh sistem menjadi 2 berdasar ukuran citra), dan threshold *region merging* di set 0,4.



| Tr-Kuan | Jml-Cluster | Jml Reg | Reg Merg | Total Komp (detik) |
|---------|-------------|---------|----------|--------------------|
| 1 | 15 | 19 | 17 | 47 |
| 64 | 12 | 15 | 15 | 52 |
| 128 | 10 | 12 | 9 | 41 |
| 255 | 9 | 11 | 8 | 54 |
| 512 | 7 | 11 | 7 | 56 |
| 1000 | 5 | 9 | 7 | 46 |
| 5000 | 2 | 5 | 2 | 29 |
| 8000 | 2 | 5 | 2 | 29 |
| 10000 | 1 | 1 | 1 | 23 |

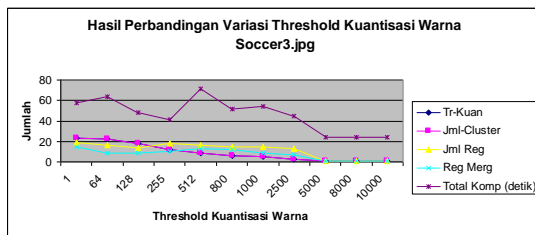


Gambar 5. Hasil Segmentasi, Tabel dan Grafik Perbandingan pada Variasi Threshold Kuantisasi Warna Golf3.jpg

Sedang hasil segmentasi pada salah satu citra *event* olah raga sepak bola golf3.jpg ditunjukkan pada Gambar 6. Set parameter yang sama, yaitu variasi nilai parameter *threshold* kuantisasi, jumlah skala 1 (oleh sistem menjadi 2 berdasar ukuran citra), dan threshold *region merging* di set 0,4



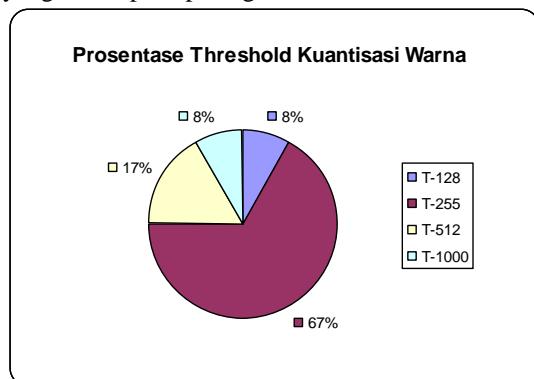
| Tr-Kuan | Jml-Cluster | Jml Reg | Reg Merg | Total Komp (detik) |
|---------|-------------|---------|----------|--------------------|
| 1 | 23 | 19 | 15 | 58 |
| 64 | 22 | 16 | 9 | 64 |
| 128 | 18 | 14 | 9 | 48 |
| 255 | 12 | 18 | 10 | 41 |
| 512 | 9 | 16 | 13 | 71 |
| 800 | 6 | 15 | 12 | 52 |
| 1000 | 5 | 15 | 9 | 54 |
| 2500 | 3 | 13 | 7 | 45 |
| 5000 | 1 | 1 | 1 | 24 |
| 8000 | 1 | 1 | 1 | 24 |
| 10000 | 1 | 1 | 1 | 24 |



Gambar 6. Hasil Segementasi, Tabel dan Grafik Perbandingan pada Variasi Threshold Kuantisasi Warna Soccer3.jpg

Dari pengamatan, semakin besar threshold kuantisasi warna, jumlah cluster warna semakin berkurang karena semakin banyak *cluster* warna yang digabungkan. Namun jumlah region yang terbentuk tidak selalu berkurang mengikuti jumlah *cluster*. Sedangkan jumlah *region merging* juga akan mengikuti dari jumlah *region* yang terbentuk. Untuk waktu komputasi juga cenderung makin berkurang.

Berdasarkan uji coba variasi nilai *threshold* kuantisasi warna pada setiap citra, hasil segmentasi citra terbaik cenderung pada nilai parameter 255 dengan prosentase 67% dibanding yang lain seperti pada gambar 7.



Gambar 7. Grafik Prosentase Threshold Kuantisasi Warna

4 Kesimpulan

Dari hasil uji coba yang dilakukan, didapatkan beberapa simpulan sebagai berikut:

1. Penentuan nilai parameter pada proses segmentasi ini sangat menentukan hasil segmentasi citra yang sesuai.
2. Parameter nilai *threshold* kuantisasi warna berpengaruh terhadap jumlah *cluster* warna yang terbentuk, dimana semakin besar nilai parameter, jumlah *cluster* warna semakin sedikit karena semakin banyak *cluster* yang digabungkan, demikian pula sebaliknya. Dan waktu komputasi juga makin kecil karena makin sedikit cluster warna yang diproses.
3. Berdasarkan uji coba pada variasi nilai *threshold* kuantisasi warna, hasil segmentasi citra terbaik cenderung pada nilai parameter 255 dengan prosentase 67% dibanding yang lain.

Dengan kombinasi parameter yang tepat, sangat berguna pada pengembangan berikutnya khususnya dalam penerapan sistem temu kembali citra dengan memanfaatkan hasil segmentasi dari algoritma JSEG.

5 Daftar Pustaka

1. D. Comaniciu and P. Meer, “*Robust analysis of feature spaces: color image segmentation*”, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp 750-755, 1997.
2. Y. Deng, B. S. Manjunath, dan H. Shin. “*Color Image Segmentation*”, In : Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR '99, Fort Collins, CO, vol. 2, pp. 446-51, Juni 1999.
3. Y. Deng, C. Kenney, M. S. Moore, dan B. S. Manjunath, “Peer Group Filtering and Perceptual Color Image Quantization”, *to appear in Proc. of ISCAS*, 1999.
4. Y. Deng, B. S. Manjunath, “Unsupervised Segmentation of Color-Texture Regions in Images and Video”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI '01)*, vol. 23, no. 8, pp. 800-810, Agustus 2001.

